

What is claimed is:

1 1. A downlink beam frame scheduler comprising:

2 a memory; and

3 a schedule table stored in the memory, the
4 schedule table comprising:

5 a scheduling segment comprising at least one
6 scheduling entry, the scheduling entry comprising a
7 header field defining at least one of a payload and
8 frame type for at least one of a payload and frame to
9 be transmitted, and payload data pointers to data in
10 memory to be transmitted in the payload.

1 2. The frame scheduler of claim 1, wherein the

2 header field defines a payload type indicative of
3 coding rate for the payload.

1 3. The frame scheduler of claim 1, wherein the
2 scheduling entry comprises a first payload scheduling
3 entry for the payload in the frame, and wherein the

- 52 -

4 scheduling table further comprises a second payload
5 scheduling entry for a second payload in the frame.

1 4. The frame scheduler of claim 1, wherein the
2 header field defines a first payload type field for a
3 first payload in the frame and a second payload type
4 field for a second payload in the frame.

1 5. The frame scheduler of claim 1, wherein the
2 payload data pointers comprise queue pointers.

1 6. The frame scheduler of claim 5, wherein the
2 queue pointers are indicative of downlink beam hop
3 location.

1 7. The frame scheduler of claim 6, wherein the
2 queue pointers are further indicative of priority.

1 8. The frame scheduler of claim 7, wherein the
2 queue pointers are further indicative of code rate.

1 9. The frame scheduler of claim 8, wherein the
2 code rate is one of a light and heavy code rate.

1 10. The frame scheduler of claim 1, wherein the
2 memory comprises a plurality of scheduling segments
3 for directing preparation of downlink frames.

1 11. The frame scheduler of claim 1, wherein the
2 payload header further defines a frame offset pointing
3 to a subsequent payload header.

1 12. The frame scheduler of claim 1, wherein the
2 header field defines a power gated payload type.

1 13. The frame scheduler of claim 1, wherein the
2 header field defines a power gated frame type.

1 14. The frame scheduler of claim 1, wherein the
2 scheduling segment comprises a plurality of scheduling
3 entries, each scheduling entry directing preparation
4 of a subsequent downlink frame.

1 15. The frame scheduler of claim 1, wherein the
2 data are ATM cells.

1 16. A downlink frame processing system for a
2 satellite, the frame processing system comprising:

- 54 -

3 a packet switch routing self addressed uplink
4 data from an input port to an output port;

5 a memory coupled to the output port, the memory
6 comprising storage for at least two downlink beam hop
7 locations; and

8 a downlink scheduler coupled to the memory, the
9 downlink scheduler including a downlink schedule
10 comprising at least one scheduling entry, the
11 scheduling entry comprising a header field defining at
12 least one of a payload and frame type for at least one
13 of a payload and frame to be transmitted, and payload
14 data pointers into the memory.

1 17. The frame processing system of claim 16,
2 wherein the downlink scheduler comprises a plurality
3 of scheduling segments, each including a downlink
4 schedule.

1 18. The frame processing system of claim 17,
2 wherein one of the scheduling segments is an active

- 55 -

3 scheduling segment, and the remaining segments are
4 inactive scheduling segments.

1 19. The frame processing system of claim 16,
2 wherein the uplink data are ATM cells.

1 20. The frame processing system of claim 16,
2 wherein the header field defines a first payload type
3 field for a first payload in the frame and a second
4 payload type field for a second payload in the frame.

1 21. The frame processing system of claim 16,
2 wherein the payload data pointers comprise queue
3 pointers.

1 22. The frame processing system of claim 20,
2 wherein the queue pointers are indicative of downlink
3 beam hop location.

1 23. The frame scheduler of claim 22, wherein the
2 queue pointers are further indicative of priority.

1 24. The frame scheduler of claim 23, wherein the
2 queue pointers are further indicative of code rate.

- 56 -

1 25. The frame scheduler of claim 24, wherein the
2 code rate is one of a light and heavy code rate.

1 26. A method for preparing downlink frames for
2 transmission in a satellite downlink, the method
3 comprising:

4 switching self addressed uplink data from a
5 switch input port to a switch output port;

6 allocating, in a memory, storage for at least two
7 downlink beam hop locations; and

8 forming downlink frames by processing a downlink
9 schedule comprising at least one scheduling entry, a
10 header field in the scheduling entry defining at least
11 one of a payload and frame type for at least one of a
12 payload and frame to be transmitted, and payload data
13 pointers into the memory.

1 27. The method of claim 26, wherein allocating
2 comprises allocating a first downlink beam hop
3 location queue and a second downlink beam hop location
4 queue.

- 57 -

1 28. The method of claim 27, wherein processing
2 comprises processing an active one of a plurality of
3 scheduling segments storing the downlink schedule.

1 29. The method of claim 28, further comprising
2 deactivating the active one of the scheduling segments
3 and activating a different scheduling segment in the
4 plurality of scheduling segments.

1 30. The method of claim 26, wherein processing
2 the payload data pointers comprises processing queue
3 pointers.

1 31. The method of claim 30, wherein processing
2 queue pointers comprises processing queue pointers
3 indicative of downlink beam hop location.

1 32. The method of claim 31, wherein processing
2 queue pointers comprises processing queue pointers
3 indicative of priority.

1 33. The method of claim 32, wherein processing
2 queue pointers comprises processing queue pointers
3 indicative of code rate.

- 58 -

1 34. The method of claim 30, further comprising
2 the step of servicing a different queue when a
3 scheduled queue indicated by a queue pointer is empty.

1 35. The method of claim 33, further comprising
2 the step of servicing a light coding queue when a
3 heavy coding queue indicated by a queue pointer is
4 empty.